

Mocking Dependencies in Spring Tests using Mockito

Exercise 1: Mocking a Service Dependency in a Controller Test

****Task:**** Write a unit test for a Spring controller that uses a service to fetch data. Mock the service dependency using Mockito.

```
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.Id;
6 @Entity
7 public class User {
8     @Id
9     private Long id;
10    private String name;
11    public Long getId() {
12        return id;
13    }
14    public void setId(Long id) {
15        this.id = id;
16    }
17    public String getName() {
18        return name;
19    }
20    public void setName(String name) {
21        this.name = name;
22    }
23 }
```

```
4 import com.example.User;
5 import org.springframework.data.jpa.repository.JpaRepository;
6
7 public interface UserRepository extends JpaRepository<User, Long> {
8 }
```

```

4  ✓ import com.example.User;
5      import com.example.UserRepository;
6      import org.springframework.beans.factory.annotation.Autowired;
7      import org.springframework.stereotype.Service;
8
9      3 usages
10     @Service
11     public class UserService {
12         1 usage
13         @Autowired
14         private UserRepository userRepository;
15
16         2 usages
17         public User getUserById(Long id) {
18             return userRepository.findById(id).orElse( other: null);
19         }
20     }

```

```

5      import org.springframework.beans.factory.annotation.Autowired;
6      import org.springframework.http.ResponseEntity;
7      import org.springframework.web.bind.annotation.*;
8
9      1 usage
10     ✓ @RestController
11     @RequestMapping("/users")
12     public class UserController {
13
14         1 usage
15         @Autowired
16         private UserService userService;
17
18         1 usage
19         @GetMapping("/{id}")
20         public ResponseEntity<User> getUser(@PathVariable Long id) {
21             return ResponseEntity.ok(userService.getUserById(id));
22         }
23     }

```

```

3  import org.junit.jupiter.api.Test;
4  import org.mockito.InjectMocks;
5  import org.mockito.Mock;
6  import static org.mockito.Mockito.when;
7  import static org.assertj.core.api.Assertions.assertThat;
8  import org.junit.jupiter.api.extension.ExtendWith;
9  import org.mockito.junit.jupiter.MockitoExtension;
10 import org.springframework.http.ResponseEntity;
11 @ExtendWith(MockitoExtension.class)
12 public class UserControllerTest {
13     @Mock
14     private UserService userService;
15     @InjectMocks
16     private UserController userController;
17     @Test
18     void testGetUser() {
19         User user = new User();
20         user.setId(1L);
21         user.setName("Alice");
22         when(userService.getUserById(1L)).thenReturn(user);
23         ResponseEntity<User> response = userController.getUser(id: 1L);
24         assertThat(response.getBody().getName()).isEqualTo(expected: "Alice");
25     }
26 }

```

✓ UserControllerTest (cor 1 sec 825 ms)

✓ Tests passed: 1 of 1 test – 1 sec 825 ms

✓ testGetUser() 1 sec 825 ms

"C:\Program Files\Java\jdk-21\bin\java.exe" ...

Exercise 2: Mocking a Repository in a Service Test

****Task:**** Write a unit test for a Spring service that uses a repository to fetch data. Mock the repository dependency using Mockito.

```
3  import org.junit.jupiter.api.Test;
4  import org.mockito.InjectMocks;
5  import org.mockito.Mock;
6  import java.util.Optional;
7  import static org.mockito.Mockito.when;
8  import static org.assertj.core.api.Assertions.assertThat;
9  import org.junit.jupiter.api.extension.ExtendWith;
10 import org.mockito.junit.jupiter.MockitoExtension;
11 @ExtendWith(MockitoExtension.class)
12 public class UserServiceTest {
13     1 usage
14     @Mock
15     private UserRepository userRepository;
16     1 usage
17     @InjectMocks
18     private UserService userService;
19     @Test
20     void testGetUserById() {
21         User user = new User();
22         user.setId(2L);
23         user.setName("Bob");
24         when(userRepository.findById(2L)).thenReturn(Optional.of(user));
25         User result = userService.getUserById(2L);
26         assertThat(result.getName()).isEqualTo("Bob");
27     }
28 }
```

✓ UserServiceTest (com.example) 3 sec 869 ms passed: 1 of 1 test – 3 sec 869 ms

✓ testGetUserById() 3 sec 869 ms

"C:\Program Files\Java\jdk-21\bin\java.exe" ...

Exercise 3: Mocking a Service Dependency in an Integration Test

****Task:**** Write an integration test for a Spring Boot application that mocks a service dependency using Mockito.

```
2  import org.junit.jupiter.api.Test;
3  import static org.mockito.Mockito.when;
4  import org.springframework.beans.factory.annotation.Autowired;
5  import org.springframework.boot.test.context.SpringBootTest;
6  import org.springframework.boot.test.mock.mockito.MockBean;
7  import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
8  import org.springframework.test.web.servlet.MockMvc;
9  import org.springframework.test.web.servlet.request.MockMvcRequestBuilders;
10 import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*;
11 @SpringBootTest
12 @AutoConfigureMockMvc
13 public class UserIntegrationTest {
14     1 usage
15     @Autowired
16     private MockMvc mockMvc;
17     1 usage
18     @MockBean
19     private UserService userService;
20     @Test
21     void testGetUserIntegration() throws Exception {
22         User user = new User();
23         user.setId(3L);
24         user.setName("Charlie");
25         when(userService.getUserById(3L)).thenReturn(user);
26         mockMvc.perform(MockMvcRequestBuilders.get(uriTemplate: "/users/3"))
27             .andExpect(status().isOk())
28             .andExpect(jsonPath(expression: "$.name").value(expectedValue: "Charlie"));
29     }
30 }
```

```
UserIntegrationTest (com.example) Tests failed: 1, ignored: 1 of 1 test
testGetUserIntegration()
Class Configuration
"C:\Program Files\Java\jdk-21\bin\java.exe" ...
12:15:28.122 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils -- Could not detect default configuration
Test ignored.
```

Same issue like JUnit Spring Tests